Motion-based Moving Object Detection and Tracking using Automatic K-means

Arghavan Keivani , Jules-Raymond Tapamo , Farzad Ghayoor

School of Engineering, University of KwaZulu-Natal, South Africa
arghavan.kayvani@gmail.com
tapamoj@ukzn.ac.za
ghayoor@ukzn.ac.za

Abstract—**Multiple objects detection and tracking are amongst the most important tasks in computer vision-based surveillance and activity recognition. This paper proposes a real-time multiple objects detection method and compares its performance with three existing methods. 'Good Features to Track' algorithm is used to extract feature points from each frame. Based on the motion-based information, feature points corresponding to moving objects are extracted from next frame. Then, the number of moving objects in each frame is determined according to their motion-based information and position, and are later clustered using the k-means algorithm. Clustering of moving objects in this paper is performed using feature vectors made of pixels' intensities, motion magnitudes, motion directions and feature point positions. In terms of accuracy and efficiency, the proposed method is shown to be highly accurate in determining the number of moving objects and also fast in tracking them in the scene.**

***Keywords*—Moving Object Detection, Object Tracking, Good Features to Track, KLT, K-means.**

I. INTRODUCTION

Tracking moving objects is a major challenge in computer vision. It has a wide variety of applications, such as object recognition, activity analysis and classification, and video surveillance. The efficiency of these applications depends on the accuracy of object detection techniques and speed of tracking method used. For instance, in a video surveillance system that aims to identify people based on their motion information, it is essential to accurately detect the moving objects and then use a robust algorithm to track them. This will make the achieved results sufficiently reliable to be used in subsequent steps of motion-based analysis.

Object tracking process can be divided into two steps; detection of moving objects in each frame and association over time of corresponding detected objects in each frame. Due to the wide spectrum of tracking moving objects applications in machine vision, many researchers in recent years have studied and presented different methods. A comprehensive survey on this subject is presented in [1], which reviews and classifies tracking methods. In [2-5] various tracking methods are discussed.

One of the methods for object detection is background subtraction, which subtracts a background model from the current frame to detect moving objects. Since background subtraction methods are computationally efficient and capable of dealing with illumination changes, noise, and multimodal backgrounds, they are used in most of the state of the art models for detection. A method discussed in [6,7] used adaptive background subtraction for real-time tracking. However, the object region is not completely recognized in these methods [1]. Hence, object detection can be done based on information obtained from a series of features of objects. These features include edge, texture, colours of objects, motion-based information, corners points and so on. Depending on the desired application, each of these features or their combination can be used. For example, colour and texture (as visual features) are used in [8] as well as edge and optical flow in [9], both work based on object contour to perform object detection and tracking. An edge feature is one of the popular features to be used, as it is simple and accurate [10]. Authors in [11] evaluated eight different edge detection methods.

One of the most common features used for detection is corner feature, as it is less sensitive to the illumination changes compared to colour feature. Corner points are used in many tracking methods and there are several corner points detection methods including Moravec corner detection algorithm [12], A combination of corner and edge detector [13], Kanade–Lucas–Tomasi (KLT) feature tracker [14,15], and SIFT (Scale Invariant Feature Transform) [16]. Authors in [17,18] evaluated the performance of feature points for tracking.

To present every moving object individually, the feature points need to be clustered. An overview of main clustering methods is provided in [19], and a comprehensive survey of this subject is presented in [20]. K-means is the simplest and most widely used algorithm for clustering. Methods proposed in [21-23] use k-means clustering method based on geometric and colour features. The key idea in this approach is to cluster the features of the moving objects and also the surrounding background. Then, the joint clusters must be removed to discriminate the target object from its surrounding background. Since this algorithm is highly sensitive to initialization of cluster centers, various methods have been proposed and their performances were evaluated [24].

This paper presents a new real-time multiple moving objects detection method through proposing an automatic k-means clustering algorithm, which produces accurate and efficient

978-1-5386-2775-4/17/$31.00 ©2017 IEEE

results. In fact, the modified k-means clustering algorithm does not require any prior information about the number of moving objects, which is the reason to be called automatic k-means. Results of different experiments show the efficiency of the proposed method in terms of running time, memory usage and accuracy. This method uses a combination of various information for clustering, hence it performs satisfactorily against different challenges of tracking. The rest of the paper is organized as follows. Section II explains the proposed method in details. Experiments and results are provided in Section III, and the conclusion is presented in Section IV.

II. Proposed Method

The proposed method uses the motion-based information to extract feature points associated with the moving objects. The clustering and assignment of the moving objects are done using the feature vectors and an automatic k-means algorithm, based on features intensity, position, motion direction, and motion magnitude. The steps of the proposed method are shown in Fig. 1 and explained in details in the following sections.

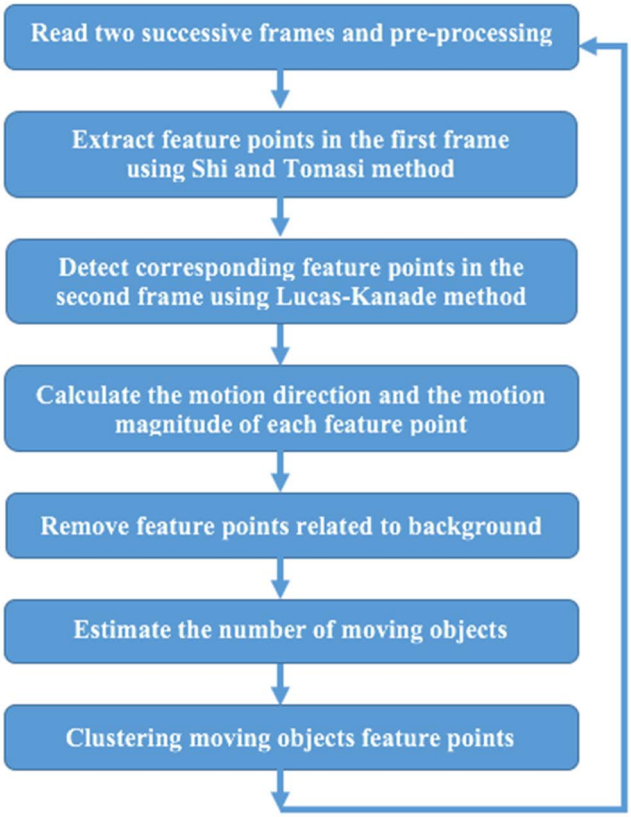


Fig. 1. The proposed method flowchart for moving objects detection and tracking in two successive frames

A. Read input frames and pre-processing

The aim of detection and tracking process is to specify the moving objects locations over time. Therefore, a set of frame sequence of moving objects is needed to locate every moving object in every frame at any time. Therefore, depending on the desired task, different situations can be considered. For example, in a surveillance system, the image sequences are received from cameras installed in specific locations. In this paper, prepared footages [25-28] are used to evaluate the proposed method.

Noise in images is inevitable as the image quality is affected by several factors. For instance, received images are affected by image acquisition devices, weather conditions, scene's texture, and illumination changes through day and night, which challenge the algorithms in computer vision. Therefore, by removing or reducing the noise, better results can be achieved for these algorithms. Pixels tend to look like their adjacent pixels. Thus, smoothing the image can predict the pixel's value from the values of its neighbouring pixels (by smoothing Gaussian noise, pixel values are no longer independent) [29]. A simple way to reduce the noise is low-pass filtering, which is integrated to edge detection for Laplacian of Gaussian method. Since we use the edge feature in our method, a Gaussian noise removal algorithm is used after receiving every frame and before any frame processing.

B. Feature Extraction

After noise removal, Shi and Tomasi algorithm [30] is used to extract feature points in frame time $t-1$, which evaluates feature points quality through frames using an affine motion model. This method uses eigenvalues of the second-moment matrix with a minimum eigenvalue threshold, to find specific feature points, introduced as 'Good Features to Track'. The key idea is that these feature points are the ones whose motion can be reliably estimated. The results of extracting these 'good feature points to track' are shown in Fig. 2, applied to frame number 50 from video 'Walking' in [26].

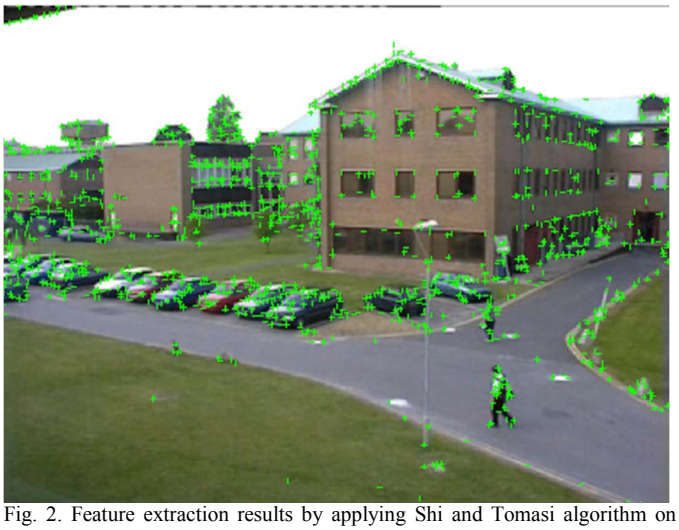


Fig. 2. Feature extraction results by applying Shi and Tomasi algorithm on frame number 50 from video 'Walking' in [26].

C. Finding Corresponding Feature Points

We use Lucas-Kanade method [14,15] to extract the corresponding feature points on the next frame (frame at time *t*). The fundamental assumptions in this method are based on temporal persistence, brightness constancy, and spatial coherence. In temporal persistence, it is assumed that points have small movement (only a few pixels) between consecutive frames. The brightness constancy constraint assumes that the projection of the particular point remains unchanged in every

frame. Lastly, based on spatial coherence constraint, points behave like their neighbours having the same properties and similar displacements. This algorithm evaluates every window of four pixels (eigenvalues by a 2×2 gradient matrix) in the frame, in terms of their texturedness property [31]. By capturing a batch of feature points from the frame at time *t-1* as input data, the result of this algorithm would be the corresponding points in the next frame (frame at time *t*). Note that some corresponding point might not exist.

Fig. 3 represents the result of applying the Lucas-Kanade method on feature points obtained from the previous step to find their corresponding points in the following frame. In Fig. 3, green plus signs illustrate the extracted feature points in frame number 50 in footage 'Walking' [26] and their corresponding points in frame number 51, which are shown in white plus signs. The extracted feature points, which are static or have very small movements, are related to the background. Therefore, some of the white plus signs overlap the green ones. The result of this step is used in the following steps of the proposed algorithm, to obtain the motion-based information of the feature points.

Fig. 3. Finding corresponding points in two consecutive frames, using the Lucas-Kanade method.

D. Computing Motion direction and Motion Magnitude of the Feature Points

The proposed method aims to detect and cluster the moving objects based on their motion-based information. Thus, it is proposed to find the motion magnitude of each feature vector as:

$$m_{i.t} = \sqrt{(x_{i.t} - x_{i.t-1})^2 + (y_{i.t} - y_{i.t-1})^2} \qquad (1)$$

and its motion direction as:

$$d_{i.t} = \arctan({}^{y_{i.t} - y_{i.t-1}}/_{x_{i.t} - x_{i.t-1}}) \qquad (2)$$

where $i = 1, \dots, n$ (n is the number of feature points that their corresponding points *are found in the subsequent frame).* $(x_{i.t-1}, y_{i.t-1})$ *and* $(x_{i.t}, y_{i.t})$ represent the coordinates of i^{th} feature point in frame time $t-1$ and its corresponding coordinate in frame time t respectively.

E. Removing the Feature Points associated with the Background

To detect the moving objects in the scene, we use the motion-based information obtained from the previous section. As the background pixels are constant and fixed, the feature points that have zero movements or only partial displacement are considered to be related to the background. In exceptional cases of minor movements like swaying trees, we consider a threshold, *T*, to cover this issue. After thresholding and removing the feature points related to the background, the remaining feature points are considered as moving objects. The formula in equation (3) categorizes the extracted feature points obtained from the previous step into two categories of the moving objects' feature points and the feature points related to the background.

$$\text{FeaturePoints} = \begin{cases} \text{MovingObjects} & \text{if } m_{i.t} > T \\ \text{Background} & \text{Otherwise} \end{cases} \qquad (3)$$

where $i = 1, \dots, n$. (n is equivalent to the number of feature points, which their corresponding points are discovered in the subsequent frame). $m_{i.t}$ represents the motion magnitude of i^{th} feature point in frame time t, obtained from formula (1) and T is the motion-based threshold. Fig. 4, represents the result of applying the formula (3) on frame number 51 from video 'Walking' in [23], showing only feature points associated with moving objects in white plus signs. In this experiment, we set the threshold as $T = 1$. In the next section, we specify the number of moving objects in a frame according to the motion-based information of the feature points obtained in this step.

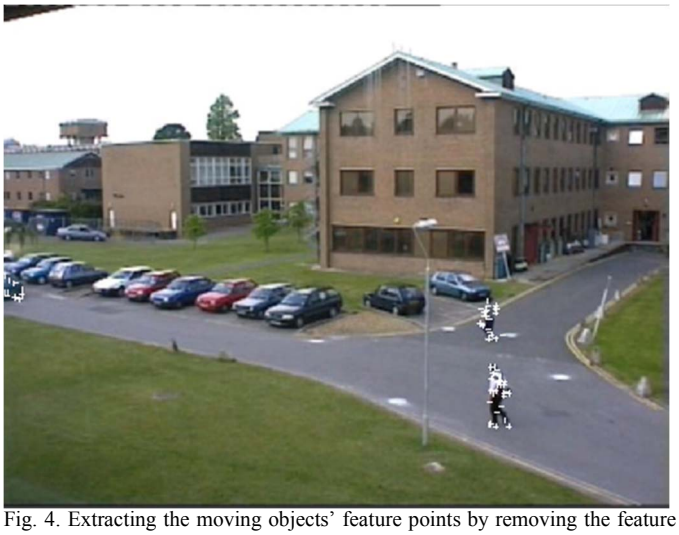


Fig. 4. Extracting the moving objects' feature points by removing the feature points associated with the background.

F. Estimation of Number of Moving Objects

K-means algorithm requires the exact number of moving objects to cluster the feature points of each moving object. Therefore, the motion-based information resulting from the previous section is used to calculate the numbers of distinct values for magnitude and direction in frame time t, based on the specified thresholds. The number of moving objects, k, will then be equal to the maximum of these two numbers, as shown

in equation (4). The feature magnitudes are converted to integer numbers, and feature directions are categorized into 24 directions (the 360 degrees of the unit circle, is divided into 24 segments). The number of moving objects is computed as

$$k = Max(m_t, d_t) \quad (4)$$

where m_t and d_t are representing the number of distinct values for magnitudes and the number of distinct values for directions in frame time t respectively. k determines the number of moving objects (clusters).

G. *Clustering the Feature Points associated with the Moving Objects with Automatic K-means*

At this step, we cluster the feature points that are associated with the moving objects, using k-means algorithm [31]. The feature vectors used for clustering the feature points include position, intensity, motion magnitude and motion direction. As the k-means algorithm needs to be initialized by the number of clusters, we use the value obtained from the previous section as the number of moving objects, k. Since in this method, the number of clusters is obtained without prior information about the number of different moving objects, we call the algorithm used in this paper "Automatic k-means Algorithm".

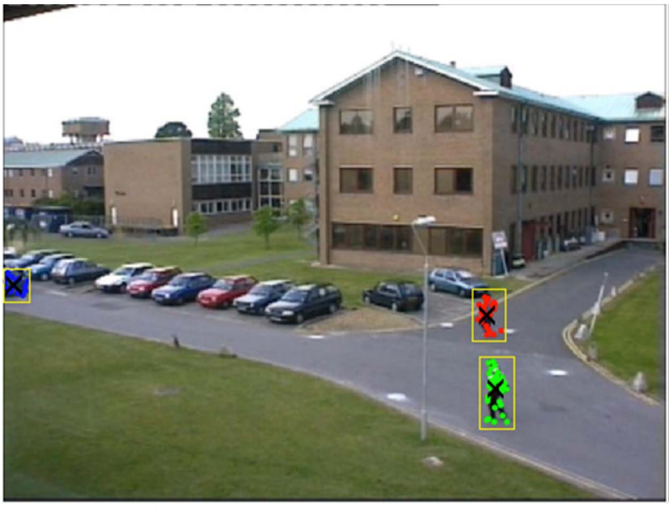


Fig. 5. Result of the automatic k-means clustering algorithm

In a case of two or more objects moving with the same magnitude but in different directions, or in the same direction but with different magnitudes, the number of clusters would be determined correctly, as we used the maximum of the number of directions and magnitudes for clustering.

There is also a case that two or more objects move in the same direction with the same magnitude. To prevent clustering these objects in the same cluster, we propose to put an extra constraint as the distance (the feature points' positions in a cluster). The algorithm checks the distance between the feature points among the same cluster, sharing the same values for both direction and magnitude. If the distance is less than the biggest magnitude of all, these feature points belong to the same cluster. But if the distance is greater than the maximum magnitude, it should be considered as a new cluster.

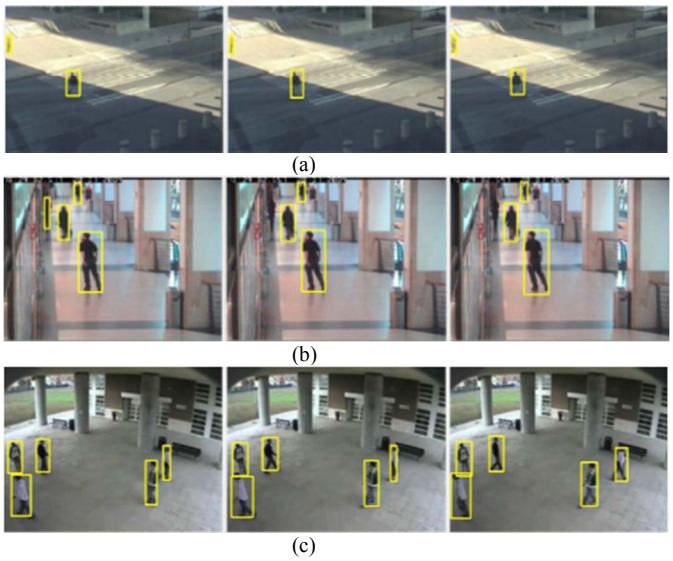
Whilst, if two moving objects move beside each other (small distance value between the two objects) with the same direction and the same magnitude, the two objects are considered to be as one. Upon these objects start moving in different directions or with different speed (moving away from each other), they would be recognized as two separate moving objects and tracked separately. Since the algorithm clusters the feature points in every two successive frames, if a feature point is classified into a wrong cluster, it would be reclassified again in the next frame.

After the clustering process, each cluster of feature points is assigned to a moving object. The mean value for each cluster is then calculated based on the feature vector and considered as the object centroid. Afterward, a rectangular shape, which is centered at the object centroid is drawn around each moving object, encircling all the feature points in that cluster. The result of this step is presented in Fig. 5, which the mean values are shown by black Xs.

III. Experiments and Results

Experiments are run using the proposed method and achieved results are compared with different methods in terms of running time and accuracy. Dataset used is chosen from videos provided in [25-28], which are widely used in tracking methods evaluation and contain different challenges for tracking algorithms. Fig. 6, displays the results of running the proposed algorithm on three different video sequences.

In this paper, we used the video 'Walking' from [26], which contains different types of objects in the scene. After running the proposed method on these videos, we compare the processing time for each frame and the tracking accuracy of our proposed algorithm with other methods provided in [32-34].



(a)

(b)

(c)

Fig. 6. Result of proposed method, (a) frames 775-777 of video 'seq01_cam1_300305_A' in [28] (objects moving in different directions), (b) frames 86-88 of video 'crossing' in [26] (illumination changes), (c) frames 258-360 of video 'walking2' in [26] (moving towards the camera and changing direction)

We used a MacBook Pro laptop with a 2.7 GHz Intel Core i7 processor and a 4GB memory with Intel HD Graphics 3000 384 MB. Accuracy and running time of the proposed method is compared with three other methods and the results are shown in Tables I, II, and III.

A. Accuracy Evaluation

To evaluate the accuracy of the proposed algorithm, we used 10 video samples, which have been used in different methods for accuracy evaluation. Based on formula (5), we evaluate the proposed method accuracy on each video. Then, the average of these 10 obtained accuracy values is calculated. Table I represents the results of proposed method accuracy on these 10 samples. A comparison of the proposed method accuracy with the other three methods is shown in Table III.

Formula (5) calculates the accuracy of the proposed method based on correctly recognized moving objects, as well as the wrong ones.

$$\text{Accuracy} = {}^{\text{CR}}/_{(\text{CR}+\text{ER})} \qquad (5)$$

where CR represents the number of moving objects, which have been correctly recognized and ER is the number of errors in detecting moving objects.

TABLE I. Result of accuracy evaluation of the proposed method

Video Name in [25-28]	Number of Frames	Number of Correct Detections	Number of False Detections	Accuracy
coke	291	268	23	92%
crossing	120	300	29	91%
Fight_RunAway2	551	765	91	88%
OneLeaveShopReenter1	395	363	36	90%
seq01_cam1_300305_A	2292	4815	289	94%
subway	176	1217	212	85%
Sylvester	1345	1171	174	87%
Tiger1	354	323	31	91%
Walking	412	822	53	94%
Walking2	500	1497	203	88%
Average Accuracy				90%

It has to be taken into consideration that in the case of new objects entering the scene, they would be recognized and tracked in the next frame since their motion-based information is obtained based on their corresponding feature points' movement obtained from the next frame. The reason behind the high accuracy of the proposed method is that it uses corner points and feature vector for clustering. This creates an adequate performance in different conditions such as illumination changes, changes of motion directions and various types of moving objects.

B. Evaluating Running Time

To evaluate the execution time and the speed performance, the proposed method was run on 10 different video samples to obtain the running time of each video sequences. Then, we divided the total running time of each video sequence by the number of its frames to get the average runtime for each frame in that specific video. Afterward, we get the average of these 10 average run-times to get the total average running time of each frame.

Table II presents results of execution time for the proposed algorithm. Results of comparison the proposed algorithm with other three algorithms, in term of running time, can be found in Table III. Since the proposed method only works with feature points associated with moving objects, its execution time is less than other methods, which process the entire frames.

TABLE II. Result of evaluating the proposed method runtime

Video Name [25-28]	Number of Frames	Total Running Time (ms)	Running Time for Each Frame (ms)
coke	291	7566	26
crossing	120	3240	27
Fight_RunAway2	551	13775	25
OneLeaveShopReenter1	395	11060	28
seq01_cam1_300305_A	2292	61884	27
subway	176	5280	30
Sylvester	1345	41695	31
Tiger1	354	9912	28
Walking	412	11948	29
Walking2	500	14500	29
Average Running Time			28

TABLE III. Comparing the accuracy and runtime of the proposed method with three different methods.

Methods	Accuracy	Running Time for Each Frame (ms)	Number of Processed Frames per second (FPS)
Method Presented in [32]	85%	40	25
Method Presented in [33]	84%	29	34
Method Presented in [34]	75%	38	26
Proposed Method	90%	28	36

IV. CONCLUSION

A new, fast and accurate method for detecting and tracking moving objects is proposed in this paper. In addition, the proposed method is not sensitive to the form of moving objects and have a good detection performance in the presence of several different types of objects in the scene. Furthermore, this method accurately estimates the number of moving objects in every frame, which is suitable in cases where this information is not available or it is difficult to obtain. Finally, since the proposed method only works with feature points, not only its execution time is less than that of other methods, but also its memory usage is low, as it does not process the entire frames.

REFERENCES

[1] A. Yilmaz, O. Javed, M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1-45, 2006.

[2] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, no. 3, Article 16. 2011.

[3] S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong, "Intelligent visual surveillance - A survey," *International Journal of Control, Automation, and Systems*, vol. 8, no. 5, pp. 926-939, 2010.

[4] V. Lepetit, P. Fua, "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1-89, Oct. 2005.

[5] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, p. 4-18, 2007.

[6] C. Stauffer, W.E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 8, pp. 747-757, Aug. 2000.

[7] R. Zhang, J. Ding, "Object tracking and detecting based on adaptive background subtraction," *International Workshop on Information and Electronics Engineering (IWIEE) Procedia Engineering*, vol. 29, no. 2012, pp. 1351-1355, 2012.

[8] A. Yilmaz, X. Li, M. Shah, "Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, 2004.

[9] M. Yokoyama, "A contour-based moving object detection and tracking", *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 271-276, 2005.

[10] J. F. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[11] K. Bowyer, C. Kranenburg, A. Dougherty, "Edge detector evaluation using empirical ROC curves," *Comput. Vis. Image Understand.*, vol. 84, no. 1, pp. 77-103, 2001.

[12] H. P. Moravec, "Visual mapping by a robot rover," *Proc. 6th Int. Joint Conf. Artificial Intell.*, pp. 598-620, 1979.

[13] C. Harris, M. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Conf.*, pp. 147-151, 1988.

[14] B. D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981.

[15] C. Tomasi and T. Kanade, "Detection and tracking of point features," *Technical Report CMU-CS-91132, Carnegie-Mellon University*, 1991.

[16] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004.

[17] K. Mikolajczyk, C. Schmid, "A Performance Evaluation of Local Descriptors," *Proc. Computer Vision and Pattern Recognition*, vol. 2, pp. 257-263, 2003.

[18] K. Mikolajczyk, C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[19] L. Rokach, O. Maimon, "Clustering methods," *in The Data Mining and Knowledge Discovery Handbook, New York, USA:Springer*, pp. 321-352, 2005.

[20] D. Xu, Y. Tian, "A Comprehensive Survey of Clustering Algorithms", *Ann. Data. Sci.*, vol. 2, no. 2, pp. 165-193, 2015.

[21] C. Hua, H. Wu, Q. Chen, T. Wada, "K-means Tracker: A General Algorithm for Tracking People," *Journal of Multimedia*, vol. 1, pp. 46-53, 2006.

[22] C. Hua, H. Wu, Q. Chen, T. Wada, "Object Tracking with Target and Background Sample," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 4, pp. 766-774, 2007.

[23] C. Hua, H. Wu, Q. Chen, T. Wada., "K-means Clustering Based Pixel-wise Object Tracking," *IPSJ Online Trans.*, vol. 3, pp. 820-833, 2008.

[24] M. Celebi, H. Kingravi, P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200-210, Sep. 2012.

[25] VIVID Tracking Evaluation Web Site. Available: http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html. last visited: 15-3-2017

[26] Visual Tracker Benchmark. Available: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html last visited: 15-3-2017

[27] CAVIAR Test Case Scenarios. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. last visited: 15-3-2017

[28] ViSOR, Video Surveillance Online Repository. Available: http://www.openvisor.org/video_categories.asp last visited: 15-3-2017

[29] D. Forsyth, J. Ponce, "Computer Vision: A Modern Approach," NJ, Upper Saddle River: Prentice-Hall, 2003, pp.219-234.

[30] J. Shi, C. Tomasi, "Good Features to Track," *Proc. Ninth IEEE Conf. Computer Vision and Pattern Recognition*, 1994.

[31] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, pp. 129-137, 1982.

[32] S. He, Q. Yang, R. W. H. Lau, J. Wang, M.-H. Yang, "Visual tracking via locality sensitive histograms," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2427-2434, 2013.

[33] K. Zhang, L. Zhang, M.-H. Yang, "Real-time compressive tracking," *Proc. 12th Eur. Conf. Comput. Vis.*, pp. 864-877, 2012.

[34] S. Hare, A. Saffari, P. H. S. Torr, "Struck: Structured output tracking with kernels," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 263-270, 2011.

Arghavan Keivani did her BSc in Iran in Computer Engineering and her MSc at the University of Surrey in the UK in Information and Business Systems Engineering. She is a Ph.D. student at the University of Kwa-Zulu Natal and her research interest is on "Tracking Moving Object in a Dynamic Environment".

Jules R Tapamo is Professor of Computer Science and Engineering at the School of Engineering, University of KwaZulu-Natal, South Africa. His research interests include Image Processing, Computer Vision, Machine Learning, Biometrics, Intelligent Monitoring, Activity Recognition, Surface Characterization and Formal Methods. He is a member of the IEEE Computer Society, IEEE Signal Processing Society, IEEE Geoscience and Remote Sensing Society, IEEE Computational Intelligence Society and the ACM.

Farzad Ghayoor did his BSc and MSc in Iran and his Ph.D. at the University of KwaZulu-Natal, South Africa in Electronic Engineering. He is currently with the School of Engineering at the University of KwaZulu-Natal and his research interests include Digital Communication and Signal Processing.